REMARKS

As a preliminary matter, Applicants wish to thank the Examiner for the notice that Claims 1 through 10 are allowed and Claims 15 and 16 would be allowed if rewritten in independent form. Claims 1-20 are pending.

Claims 11-14 and 17-20 currently stand rejected under 35 U.S.C. § 103(a) as being anticipated by U.S. Patent No. 5,850,323 to Engstrom *et al.* (Engstrom). Support for newly added Claims 21-24 may be found in original Claims 1-20 and in the specification on page 4, among other places.

Engstrom

Engstrom provides a method for flipping images in a window using overlays. (Engstrom, ¶2, lines 21-64.) Engstrom teaches the well-known technique of screen flipping:

> In computer generated graphics, a technique known as "screen flipping" is commonly used to provide smooth animation. In this technique, **two memory buffers** in video memory are used to generate an image. While **a first image is being rendered to a first buffer**, the display hardware scans out a complete image from a second buffer. To update the display with a new image, the display hardware then performs a buffer swap. The display image that was just under construction is then transferred to the display screen, and a new image is constructed in the buffer that held the previous display image.

(Engstrom, Col. 1, lines 40-50) (emphasis added). Engstrom, therefore, is limited to using two memory buffers for rendering to a first buffer and scanning out of a second buffer. The support for flipping images in a window is implemented in a software interface for a display device in a computer. *Id.* In this context, screen flipping enables application programs to flip in a window without disturbing other parts of the display image. *Id.*

> When an application makes a call to modify a surface, for example, the display device interface makes sure that it is safe to modify the underlying surface memory. (¶19, lines 6-9). However, since **a hardware page flip is not involved**, the flip

> control does not have to ensure that the display controller has completed a previous page flip request. (¶19, lines 43-45).
>
> Unfortunately, **most display hardware does not specify explicitly** when it is safe to draw to a back buffer, or in other words, when it has completed reading these addresses. As such, the display device interface (in conjunction with the HAL or display driver on the host PC) has to determine when it is safe to: 1) modify a back buffer in response to a flip, bit, or lock request; and 2) in the case of a flip request, alter the display address. (¶ 20, lines 37-40) (emphasis added).
>
> Another check, shown in dashed lines (454) in FIG. 12A, is to check whether the hardware explicitly indicates that it has completed a **page** flip. (¶ 21, lines 49-51) (emphasis added).

Accordingly, Engstrom teaches that display hardware does not specify explicitly when it is safe to draw to the back buffer and therefore teaches away from the rejected claims. To the extent that Engstrom describes a hardware mechanism for managing access to the frame buffer, Engstrom is limited to hardware that indicates whether a page flip has occurred, not "prohibiting, by a write behind controller in a video graphics adapter, a second portion of the image primitive from being stored to a second portion of the frame buffer after the step of storing the first portion, wherein the second portion of the frame buffer is adjacent to the first portion of the frame buffer." Further, Engstrom teaches:

> Next, the driver compares the current time with the sum of the time of the last flip request and the refresh time (414). If an entire refresh period has not elapsed since the last flip request, it is not safe to change the state of the display controller. As such, the driver returns a "WasStillDrawing" error (416). (¶ 20, lines 58-63).
>
> If a refresh period has elapsed since the last flip request, the driver records the current time of the flip request and proceeds to update the hardware register (418 and 420). Specifically, the driver writes the address of the surface memory of the new front buffer to the display address. At this point, the driver has successfully completed the flip and it returns. (¶ 20, lines 64-67, ¶21, lines 1-3).

> Specifically, the driver checks the current time and determines whether a refresh period has elapsed since the last flip. If not, the error is returned. (¶21, lines 10-11).
>
> In addition, or as an alternative to using the time of the last flip request, the driver can evaluate whether it is safe to complete a flip by determining if the display controller has moved outside the VB period since the last flip request. If the display controller is not in the VB period, but has entered it since the previous flip was initiated, it is safe **to assume** the flip has completed and the display address has been changed. (¶21, lines 14-24). (emphasis added).
>
> Another optimization in the flip control is to read the scan line register, and analyze the scan line position relative to the position **when the last flip occurred.** If the scan line is less than the scan line at the time the last flip occurred, then it **is safe to assume** the previous flip operation has completed and the display address has been changed.

(Engstrom, Col. 21, lines 28-32) (emphasis added). Accordingly, Engstrom is limited to waiting until it is safe to assume that a flip has completed, based on waiting for an entire refresh period to have elapsed since the last flip request, rather than "prohibiting, by a write behind controller in a video graphics adapter, a second portion of the image primitive from being stored to a second portion of the frame buffer after the step of storing the first portion, wherein the second portion of the frame buffer is adjacent to the first portion of the frame buffer."

The teachings of Engstrom appear to be similar to the prior art system described in Applicants' Background of the Invention section. Prior art systems assume the primitive will not cause tearing because *it is necessary to wait until the display engine indicates all locations of the frame buffer needed to store the primitive image have been rastered.* (Specification, page 2, lines 24-37.) Tearing occurs when the rendering engine writes new image information into the frame buffer over frame buffer locations that contain data yet to be displayed from the previous image. (Specification, page 2, lines 2-6.) In other words, during a single screen refresh cycle, portions of data from two frames of data will be displayed. *Id.* Tearing produces non-

contiguous images that are detectable by users. *Id.* As a result, Engstrom teaches the prior art solution discussed in the prior art section of the specification, namely, waiting until the display engine indicates all locations of the frame buffer needed to store the primitive image have been rastered. Since the claimed invention stores the image after the raster has accessed data in the first memory location, the claimed invention can write to the frame buffer much earlier than with the system in Engstrom. Consequently, the Engstrom system requires a greater amount of memory than the claimed invention because it is necessary to wait until the display engine indicates that all locations of the frame buffer needed to store the primitive image have been rastered. As a result, Engstrom requires the extra memory space of both a front buffer and a back buffer, as is typical in flipping structures, and as acknowledged in the office action at page 2, reference number 5.

As an example of the advantages of the claimed invention, as described in the specification, where a large triangle is to be issued for rendering, and only a small portion of the triangle is below the line currently being rastered, the prior art (such as Engstrom) operation can result in the display engine indicating the frame buffer is not ready. Therefore, a dispatch of the operation is stalled in the prior art system even though the rendering engine could be doing useful work on most of the triangle. (See, for example, Specification, page 2, lines 22-30, page 3, line 1.)

Independent Claim 11

The write behind controller receives control information from a display device controller in order to determine a current location available in a frame buffer for receiving information. (Specification, page 4, lines 1-3.) As a result, the write behind controller may store a first portion of an image primitive to the first portion of the frame buffer "after the step of displaying the first portion of video/graphics data," instead of waiting until it is safe to assume that a flip

has completed and the buffer is full, as required by Engstrom. Therefore, the claimed write behind controller solves the problem of determining the earliest point in time that data can be written into the frame buffer so that the frame buffer and the rendering engine will be used most efficiently, thereby requiring less memory and reducing rendering and display time delays.

The cited portions of Engstrom teach that if the scan line is less than the scan line at the time the last flip occurred, then it is safe to assume the previous flip operation is completed. Engstrom, therefore, is referring to a completely different parameter or condition for writing into the frame buffer than the claimed second memory location for indicating the raster has accessed data at the first memory location. For example, the claimed write behind controller may have accessed data at the first memory location, but a flip may not have occurred because the raster has not finished reading the entire display image in the frame buffer. Because Engstrom requires that the rendering engine must wait for the raster engine to finish reading the entire display buffer and request a flip, the graphics processor is idle and not performing any useful work during this time period. For example, the graphics processor may be writing to the frame buffer locations previously rastered before the raster has finished reading from the frame buffer. As a result, Engstrom teaches writing into the frame buffer based only on a different condition, namely, only after a flip has occurred rather than "after the step of displaying the first portion of the video/graphics data."

Engstrom teaches scanning without regard to the memory location since scanning is performed on an entire screen. Therefore, Engstrom is limited to "analyz[ing] the scan line position relative to the position **when the last flip occurred,**" (Col. 21, lines 28-33) (emphasis added) as opposed to "storing a first portion of an image primitive to the first portion of the frame buffer after the step of displaying the first portion of video/graphics data." In contrast to

the claims, Engstrom merely assumes that a scan line may be scanned if the scan line is less than the scan line at the time the last flip occurred. Additionally, storage of the image, as recited in the claims, occurs at a specific condition, namely, "after the step of displaying the first portion of video/graphics data."

Unlike Engstrom, the currently rejected claims are a hardware implementation to control whether or not the rendering engine can write into a specific memory location. (Claims 11 and 13; also see the specification, page 5, lines 9-11.) In contrast, Engstrom and the prior art make the determination at the system level, based on worst-case scenarios, after the entire frame buffer is rastered, by assuming it is safe to write into the buffer by waiting an entire refresh period. (Specification, page 5, lines 11-12.) As previously stated, one advantage provided by the claimed invention is that it is possible for a primitive, when received by the rendering engine, to be processed by the rendering engine and written into available locations of the frame buffer without the system having to be concerned whether a sent rendering command can be currently displayed after the entire frame buff is rastered. (Specification, page 5, lines 12-17.) The claimed invention therefore solves the problem of utilizing the frame buffer and the rendering engine more efficiently than the prior art. (Specification, page 5, lines 18-19.)

ENGSTROM DOES NOT DESCRIBE AT LEAST "PROHIBITING, BY A WRITE BEHIND CONTROLLER IN A VIDEO GRAPHICS ADAPTER, A SECOND PORTION OF THE IMAGE PRIMITIVE FROM BEING STORED TO A SECOND PORTION OF THE FRAME BUFFER AFTER THE STEP OF STORING THE FIRST PORTION, WHEREIN THE SECOND PORTION OF THE FRAME BUFFER IS ADJACENT TO THE FIRST PORTION OF THE FRAME BUFFER"

The Applicants cannot find where the Office Action lists a limitation-by-limitation reference to Engstrom for each and every limitation of the rejected claims. As a result, it is not clear what part of the recited language in Engstrom corresponds to the claim limitations. For example, nowhere can the undersigned find where the Office Action states that Engstrom recites

"a second portion of the frame buffer after the step of storing the first portion, wherein the second portion of the frame buffer is adjacent to the first portion of the frame buffer." As a result, the cited portions of Engstrom do not teach each of the claim elements arranged, as required by the claim.

As previously stated, Engstrom is limited to "analyz[ing] the scan line position relative to the position **when the last flip occurred**" as opposed to "prohibiting, by a write behind controller in a video graphics adapter, a second portion of the image primitive from being stored to a second portion of the frame buffer after the step of storing the first portion, wherein the second portion of the frame buffer is adjacent to the first portion of the frame buffer." (Emphasis added.) Instead, Engstrom teaches, "If an entire refresh period has not elapsed since the last flip request, it is not safe to change the state of the display controller." Accordingly, Engstrom does not change the state of the display controller based on "if an entire refresh period has not elapsed" in contrast to the condition as explicitly recited in the claims of "after the step of storing the first portion." Engstrom, therefore, is referring to a completely different condition for not changing the state of the display controller, namely "if an entire refresh period has not elapsed" as opposed to "prohibiting, by a right behind controller in a video graphics adaptor, a second portion of the image primitive from being stored to a second portion of the frame buffer" based on the explicitly claimed condition of "after the step of storing the first portion." For example, if the raster has accessed data at the first memory location (*i.e.*, a pixel within the display image), but the display is not flipped, or if the refresh period has not elapsed, Engstrom would prevent writing to the frame buffer, whereas the claimed invention would allow writing to the frame buffer since the raster has already displayed the first memory location. According to the rejected claims, since the first memory location is already rasterized, the graphics processor may be

performing useful work by writing the second portion to the frame buffer. However, according to Engstrom, the graphics processor would be idle while it is waiting for a flip or a screen refresh and, therefore, the graphics processor in Engstrom is waiting rather than performing useful work. Claim 11 prohibits a second portion of the image primitive from being stored to a second portion of the frame buffer "after the step of storing the first portion." Accordingly, Engstrom does not describe "prohibiting a second portion of the image primitive from being stored to a second portion of the frame buffer after the step of storing the first portion, wherein the second portion of the frame buffer is adjacent to the first portion of the frame buffer."

The Office Action acknowledges that Engstrom does not teach a video graphics adapter. (Office Action, page 4, ¶ 7.) It is well established that to establish *prima facie* obviousness, all the claim limitations must be taught or suggested by the prior art. In addition, there must be some teaching, motivation or suggestion in either the prior art or the references themselves to modify the prior art reference.

In reviewing the Office Action, the Office Action asserts "it would have been obvious to one of ordinary skill in the art at the time the invention was made to have implemented the video card as [the] video graphics adapter in order to support a de facto standard." With regard to the Examiner's assertion of the motivation of one skilled in the art to modify the system of Engstrom, a careful examination of Engstrom, as cited, reveals that Engstrom, rather than teaching "prohibiting, by a write behind controller in a video graphics adapter, a second portion of the image primitive from being stored to a second portion of the frame buffer after the step of storing the first portion, wherein the second portion of the frame buffer is adjacent to the first portion of the frame buffer," instead teaches "analyz[ing] the second line position relative to the position when the last flip occurred," such that the rendering engine in Engstrom must wait for

the raster engine to finish reading the entire display buffer and requesting a flip, thus causing the graphics processor to be idle and not perform any useful work during this time period.

Since Engstrom teaches writing to the frame buffer only after a flip operation, Engstrom teaches away from "storing a first portion of an image at the first portion of the frame buffer after the step of displaying the first portion of video/graphics data." Since Engstrom teaches away from the invention, there is no motivation to modify Engstrom and, therefore, the Office Action fails to establish a *prima facie* case of obviousness.[1] Further, such a modification would render Engstrom inoperable[2] because Engstrom teaches away from using the requisite hardware to determine if a specific location in the frame buffer has been rasterized before a flip operation, since Engstrom merely assumes that the location has been rasterized only after a flip operation. As a result, the basis for the Office Action's assertion for the source of motivation is directly opposite from the teachings of Engstrom. Further, modifying Engstrom to write to the frame buffer without waiting to analyze the scan line position relative to the position when the last flip occurred would change the principle of operation of Engstrom.[3] Consequently, the Office Action fails to establish a *prima facie* case of obviousness.

Further, Engstrom teaches away from the claims because Engstrom teaches, "If the current position of the scan line is below the previous position, **then the scan line test is inconclusive**. . . ." (Engstrom Col. 22, lines 27-30) (emphasis added). In contrast, the claims

[1] A prior art reference must be considered in its entirety, *i.e.*, as a whole including portions that would lead away from the claimed invention. (W.L. Gore & Associates, Inc. v. Garlock, Inc., 721 F.2d 1540, 220 U.S.P.Q. 303 (Fed. Cir. 1983), *cert. denied*, 469 U.S. 851 (1984) M.P.E.P. 2141.02).

[2] The proposed modification cannot render the prior art unsatisfactory for its intended purpose. In re Gordon, 733 F.2d 200, 221 U.S.P.Q. 1125 (Fed. Cir. 1984), M.P.E.P. 2143.01.

[3] If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. In re Ratti, 270 F.2d 810, 123 U.S.P.Q. 349 (C.C.P.A. 1959). *See* M.P.E.P. 2143.01.

teach a conclusive operation, namely "prohibiting, by a write behind controller in a video graphics adapter a second portion of the image primitive from being stored to a second portion of the frame buffer after the step of storing the first portion, wherein the second portion of the frame buffer is adjacent to the first portion of the frame buffer."

If the display engine has read from the first memory location in the frame buffer, and if the display has not yet flipped because the display engine is still reading from memory locations after the first memory location but before the end of the display image, then Engstrom would prevent writing to the first memory location, whereas the claims would allow writing to the first memory location. As a result, Engstrom teaches away from "prohibiting a second portion of the image primitive from being stored to a second portion of the frame buffer after the step of storing the first portion, wherein the second portion of the frame buffer is adjacent to the first portion of the frame buffer."

Accordingly, Engstrom, as cited in the Office Action, teaches away from the claims. Therefore, there is no motivation to modify Engstrom to produce the claimed invention. For at least the reasons provided above, the claims are not obvious in view of Engstrom. Accordingly, the claims are believed to be in condition for allowance.

Dependent Claim 12

As to Claim 12, Applicants respectfully reassert the above comments and submit that this claim adds additional novel and non-obvious subject matter. As previously stated, Engstrom teaches that "the flip control gets the current time and compares it with the sum of the last flip request time plus the refresh time," rather than "after the step of accessing the second portion of video/graphics data." As a result, Engstrom teaches a completely different condition than the condition explicitly recited in Claim 12. Further, because Claim 12 depends on Claim 11, Claim 12 is allowable for at least the reasons Claim 11 is allowable.

Independent Claim 13 and Dependent Claim 14

As for Claims 13 and 14, Col. 20, lines 9-7 of Engstrom have been cited which state "to avoid modifying surface memory that the display controller is reading, the display device interface (or its HAL) checks the state of the display hardware before attempting operations that could cause a conflict such as a flip, a BLT or request to lock a surface. In the case of a flip operation on a visible flipping structure, it is important to determine whether it is safe to change the address of the surface memory region that is currently serving as the front buffer" which is limited to as previously described waiting until each and every memory location has been displayed before rendering into any of the memory locations rather than "storing the second portion of the image primitive to the second portion of the frame buffer after the step of accessing the second portion of video/graphics data." As a result, since Engstrom teaches the condition of waiting until each and every memory location has been displayed by, for example, waiting until the entire refresh period has elapsed in order to assume that each and every memory location has been displayed, the claims explicitly recite the condition "after the step of accessing the second portion of video/graphics data." As such, the Office Action has ignored a limitation of the claims, namely the condition "after the step of accessing the second portion of video/graphics data." As a result, the Office Action fails to show how Engstrom teaches each and every element as arranged in the claims.

Additionally, the Office Action on page 4, ¶7 merely cites to various portions of Engstrom without showing a limitation-by-limitation description of where each and every element as arranged in the claims as taught by Engstrom. As a result, the Office Action fails to show how Engstrom teaches each and every element as arranged in the claims. Applicants respectfully reassert the relevant remarks made with respect to Claim 11. Further, Engstrom, as previously stated, teaches away from the claims and, as such, for at least these reasons, these

claims are not obvious. Furthermore, Claims 13 and 14 add additional novel and non-obvious subject matter.

Independent Claim 17, and Dependent Claims 18, 19 and 20

Claims 17-20 stand rejected, based on the same rationale for Claim 11. Applicants respectfully reassert at least the relevant remarks made above with respect to Claim 11. Accordingly, these claims are also believed to be in condition for allowance. Applicants respectfully submit that Claims 17-20 add additional novel and non-obvious subject matter, and are allowable as at least depending from an allowable base claim.

New independent Claim 21 and Claim 22

In contrast to new claims 21 and 22, Engstrom assumes that a scan line may be scanned if the scan line is less than the scan line at the time the last flip occurred. Again, Engstrom teaches the prior art method, as taught in the background section of the specification, namely, determining when it is safe to write to the frame buffer based on a worst-case scenario, *i.e.*, last page flip plus refresh time. Accordingly, the above relevant remarks are repeated. Additionally, "storing of a first portion of an image primitive," as recited in the claims, occurs at a specific condition, namely, "in response to accessing the first portion of video/graphics data for display." Further, "prohibiting, by a write behind controller in a video graphics adapter, a second portion of the image primitive from being stored to a second portion of the frame buffer" occurs until a specific condition, namely "until the first portion of the image primitive is displayed." Engstrom, however, fails to teach either of these condition becauses Engstrom waits until "it is safe to assume the previous flip operation has completed." Accordingly, Engstrom does not teach each and every element of claim 21.

Further, because Claim 22 depends on Claim 21, Claim 22 is allowable for at least the reasons Claim 21 is allowable. Applicants respectfully submit that Claim 22 adds additional

novel and non-obvious subject matter, and is allowable as at least depending from an allowable base claim. Consequently, Engstrom and the cited references do not teach all the elements as arranged in the claims 21 and 22.

New independent Claim 23 and new dependent Claim 24

Applicants repeat the above relevant remarks. In contrast to Engstrom, new claims 23 and 24, recite "prohibit write access to the plurality of memory locations" occurs at a specific condition, namely, "when at least one of the memory locations has not been displayed,." Engstrom, however, fails to teach this condition because Engstrom waits until "it is safe to assume the previous flip operation has completed." Accordingly, Engstrom does not teach "a write behind raster controller in a video graphics adapter coupled to the rendering engine to prohibit write access to the plurality of memory locations when at least one of the memory locations has not been displayed." Consequently, Engstrom and the cited references do not teach all the elements as arranged in the claims.

Further, because Claim 24 depends on Claim 23, Claim 24 is allowable for at least the reasons Claim 23 is allowable. Applicants respectfully submit that Claim 24 adds additional novel and non-obvious subject matter, and is allowable as at least depending from an allowable base claim. Consequently, Engstrom and the cited references do not teach all the elements as arranged in the claims 23 and 24.

CONCLUSION

Applicants respectfully submit that the claims are in condition for allowance, and an early Notice of Allowance is earnestly solicited. The Examiner is invited to telephone the below-listed

attorney at 312-609-7970 if the Examiner believes that a telephone conference will expedite the prosecution of the application.

Dated: February 18, 2004

Respectfully submitted,

By: _____
Themi Anagnos
Registration No. 47,388

Vedder, Price, Kaufman & Kammholz, P.C.
222 N. LaSalle Street
Suite 2600
Chicago, Illinois 60601-1003
Phone: (312) 609-7970
Facsimile: (312) 609-5005